REMARKS/ARGUMENTS

Applicant respectfully responds to the Office Action of October 27, 2004 and requests reconsideration of the application. The shortened statutory period has been extended to run through April 27, 2004.

Claims 1-43 are now pending, a total of 43 claims. Claims 1-30 are allowed, and claim 34 is indicated as allowable. Of the claims not allowed, only claims 31 and 38 are independent.

I. Claim 31

Claim 31 is discussed in paragraph 4 of the Action of October 2004, and paragraph 7 of the Action of March 2004, in the context of Hammond '686 alone. Claim 31 recites as follows:

> 31. A method, comprising the steps of:
>
> in response to an exception raised while executing a program coded in instructions of a first instruction set architecture, initiating an execution thread under an operating system coded in instructions of a second instruction set architecture;
>
> delivering the exception to the initiated thread for handling by the operating system.

Claim 31 recites a "thread," a well-established term of art. The Office Actions compare "initiating an execution thread" to elements "500a-i" of Hammond '686, and assert that a "thread" is identical to (in a § 102 sense) to a "handler" in the Hammond '686 reference. Equating these two terms is incorrect, in a number of respects.

Hammond '686 discusses "handlers" 500 at col. 9, lines 52-col. 10, line 8, and at col. 10, lines 43-65. A conventional "handler" operates as follows. When a program is interrupted, execution is transferred to the handler. Almost by definition, when a program raises an exception and invokes a handler, the part of the program that raised the exception must not be allowed to execute until the handler for that exception completes. Leaving room for rare or artificially-constructed exceptions, if an excepted program is allowed to resume execution before the handler for the exception completes, the result will be incorrect execution. Therefore, conventionally, any possibility of prematurely executing the program is removed, by executing the handler in the same thread as the program itself. Because the original context of the interrupted program is replaced with the context of the handler, the interrupted program is prevented from further execution until the handler completes. When the handler completes its

job of resolving the interrupt condition, it gives up control of the thread. Then, the scheduler picks a program to resume. The scheduler eventually picks the originally interrupted program. Thus, need and convenience nicely align in this conventional approach. Hammond '686 says nothing that would suggest that he abandoned the advantages of this conventional approach.

Claim 31 and Hammond '686 contrast in two ways. First, claim 31 requires a thread that "handles" the exception. Hammond '686 discusses no handler that executes "independently" of the program, and therefore no "thread." Second, claim 31 requires two threads to be concurrently extant – the interrupted "program," and the initiated "thread" that handles the exception.[1] Hammond '686 never mentions the word "thread" at all, let alone more than one.

Claim 31 is patentable over Hammond '686.

A. The "Yahoo!" Definition of "Thread"

As noted in Applicant's paper of July 2004, the "thread" of claim 31 is not identical to a "handler" of Hammond '686. In reply, the Office Action of October 2004 correctly quotes the Yahoo! education web site[2] for the term "thread:" "a portion of a program that can run independently [of] other portions of the program," but then ignores the words that were quoted.

As noted above, Hammond '686 at most suggests running the handler in the identical thread that was interrupted. The only "other portion of the program" mentioned in Hammond '686, that is, the portion that was interrupted, cannot even be considered for scheduling until the handler comples. Hammond's "handler" is not "independent" of the interrupted portion of the program. Without "independence," Hammond's "handler" cannot be a "thread."

By the Examiner's own choice of definition, claim 31 recites a "thread" that distinguishes Hammond '686.

---

[1] To clarify the record for purposes of any future litigation: as a practical matter, in most cases, the thread for the interrupted process must not be allowed to resume execution until the handler completes. Claim 31 requires only that a handler "thread" be "initiated," not necessarily that the interrupted thread and the handler thread be allowed to actually execute concurrently.

[2] Yahoo! Education does not provide a fully correct "definition" that precisely sets forth all of the characteristics that those of ordinary skill use to distinguish "threads" from similar related concepts. Rather, the Yahoo! definition is more a non-specialist's introductory overview of the general concepts embodied in the term.

B. A "Thread" Is Not Anything That "Handles an Interrupt"

The October 2004 Office Action, at page 2, line 26 to page 3, line 1, suggests that the term "thread" is so broad that it encompasses anything for "handling an interrupt." This paragraph of the Office Action relies on an incorrect legal test, and thereby reaches an impermissibly-broad definition of "thread." When the correct legal test is applied, the claim language distinguishes Hammond '686.

1. The Applicable Legal Test For Claim Interpretation

The Director of the U.S. Patent and Trademark Office instructs as follows, at MPEP § 2111 (bold in original, underline added):

> **2111 Claim Interpretation; Broadest Reasonable Interpretation [R-1]**
>
> **CLAIMS MUST BE GIVEN THEIR BROADEST REASONABLE INTERPRETATION**
>
> During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification."...
>
> The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach....
>
> **II. "PLAIN MEANING" REFERS TO THE ORDINARY AND CUSTOMARY MEANING GIVEN TO THE TERM BY THOSE OF ORDINARY SKILL IN THE ART**
>
> ... If extrinsic reference sources, such as dictionaries, evidence more than one definition for the term, the intrinsic record must be consulted to identify which of the different possible definitions is most consistent with applicant's use of the terms.

The Board of Patent Appeals and Interferences has noted that dictionary definitions are not applicable when they are "unreasonable" or not "consistent with the specification." *E.g.*, *Ex parte Hollingsworth*, App. No. 96-2862, www.uspto.gov/web/offices/dcom/bpai/decisions/fd962862.pdf at 10-11 (BPAI 1996) (examiner's reliance on dictionary is "indiscriminate," "absurd," and "beyond all reason;" Board reminds that a claim must be interpreted "in a manner consistent with the specification and construed as those skilled in the art would construe them").

At page 2, lines 9-10 of the Action of October 2004, the Examiner disagrees with the Director and with and Board of Patent Appeals. Instead of following their instructions, the

Examiner coins a totally new test for claim interpretation: "any conventionally accepted definition," without regard for "reasonable," "consistent with the specification," or "consistent with the interpretation [of] those skilled in the art."

When the wrong legal test is applied at the very first step in analyzing a claim, none of the rest of the Office Action can be relied upon. When an examiner exceeds the Director's instructions and acts outside the authority delegated by the Director, no rejection exists.

2. The "Interpretation That Those Skilled In The Art Would Reach"

The term "thread" is a well-established term of art in the computer sciences. Without getting into technical esoterics of a precise definition, all those skilled in the art would agree with at least the Yahoo! quote selected by the Examiner, that the notion of "thread" embodies "independence." No one of ordinary skill would consider an "exception handler" and a "thread" to be the same thing. They are as different as a yellow highlighter mark tracing a route on a map (one possible analogy to a "thread") and a garage for repairing a breakdown (analogous to a "handler"). A conventional "thread" is the mechanism by which computer systems manage concurrent execution. A conventional "handler" is invoked in a manner that prevents concurrent execution.

Any assertion that the two terms are somehow interchangeable is wrong.

3. Equating "Thread" And "Handler" Is Inconsistent With "The Interpretation That Those Skilled in the Art would Reach"

An explanation of the term "thread," and the history of the development of threads, appears in a series of lecture notes for an operating systems course at the University of California at San Diego, available at www.cs.ucsd.edu/classes/fa00/cse120/lectures/5-threads.pdf. The UCSD notes explain that the term "thread" connotes the "unit of scheduling." In contrast, a traditional handler, such as Hammond's, preempts the program's own thread, and is scheduled as a substitute for the program. Hammond's handler inhabits the same "unit of scheduling," and is therefore not an independent "thread" as recited in claim 31.

Another explanation of threads is stated at http://www.wordiq.com/definition/Thread_%28computer_science%29, enclosed with the previous Response to Office Action. In relevant part, excerpted as follows (bold in original, underline added)

> Many programming languages, operating systems, and other software development environments support what are called "**threads**" of execution. Threads are similar to processes, in that both represent a single sequence of instructions executed in parallel with other sequences, either by time slicing or multiprocessing. ...
>
> An advantage of a multi-threaded program is that it can operate faster on computer systems that have multiple CPUs, or across a cluster of machines. This is because the threads of the program naturally lend themselves for truly concurrent execution....

This explanation is consistent with the others cited here, focusing on "parallel" and "concurrent" execution, "either by time slicing or multiprocessing." The attempt to read these elements out of the definition of "thread" at pages 2-3 of the Office Action is inconsistent with the ordinary use of the term.

Another reasonable explanation appears at http://www.calpoly.edu/cgi-bin/man-cgi?pthread+3:

> A *thread* is an independent flow of control within a process, composed of a context (which includes a register set and a program counter) and a sequence of instructions to execute. ...
>
> All processes consist of at least one thread. Multi-threaded processes contain several threads. ...

This explanation is consistent with the others cited in this paper, focusing on "independent flow of control..." This "independence" is absent from Hammond '686, and therefore Hammond '686 does not use threads.

These several explanations of "thread" are all consistent with the use of the term "thread" by those in the art, and the explanation from the Yahoo! Education dictionary cited in the Office Action.[3]

4. The Definition Of "Thread" Stated At Pages 2-3 Is Not "Reasonable"

In the paragraph spanning pages 2-3, the Office Action extrapolates a definition of "thread" that is broad enough to cover any "process that is part of a larger process or program."[4]

---

[3] Note that these definitions allow for single-thread processes, in the simplest case.

[4] The Microsoft Dictionary definition is clearly not intended to be a precise "definition" for specialists; it is clearly intended only to be a "rough idea" for non-specialists. It cannot be relied upon for what it was not intended to be.

This definition is beyond all reason: it includes everything from a single instruction, to a conventional subroutine, to the purely hardware "process" of changing a transistor state from 0 to 1, to loading the operating system into memory on cold boot before the operating system's thread mechanisms are even initialized, to any mechanism for handling an interrupt.

This part of the Office Action cannot even be reconciled with the preceding paragraph of the Action itself, let alone the "reasonable" understanding of one of ordinary skill.

To the extent that the Office Action suggests (e.g., paragraph at page 2, lines 5-7) that the claims should be redrafted to "limit the definition of thread" to only the ordinary meaning of the word "thread" but to exclude the meanings extrapolated in the Office Action, the Office Action asks the impossible. To the knowledge of the undersigned, there is no other term that has a recognized definition that is the same as the recognized definition for the term "thread." If the meaning of the word "thread" is distorted as proposed in the Office Action, there is no other word available to take its place.

A claim may not be rejected based on a definition that is not "reasonable."

5. Equating "Thread" and "Handler" is Inconsistent with the Specification

The specification uses the term "thread" and the term "handler" in their ordinary senses, and uses them to describe different things. Often, the very same sentence uses both terms together. *E.g.*, page 5, lines 20-22; page 7, line 31-page 8, line 4; page 64, lines 4-5; page 66, lines 8-13; page 72, lines 6-8; page 75, lines 17-18. If the two words are asserted to be identical, then these sentences make no sense. Equating them is "inconsistent with the specification," and thus not permissible.

C. Conclusion

For these reasons, the "thread" of claim 31 is not found in Hammond '686. Any rejection may be withdrawn.

II. Claims 32, 33, 35 and 37-43

Claim 38 recites language similar to claim 31, and is patentable for similar reasons.

Claims 32, 33, 35, 37, and 39-43 are dependent on the independent claims discussed above, and patentable therewith. In addition, the dependent claims recite additional features that further distinguish the art.

III. Inventorship

Several of the inventors are no longer easily contacted. Appropriate papers to correct inventorship will be filed when the necessary information can be gathered.

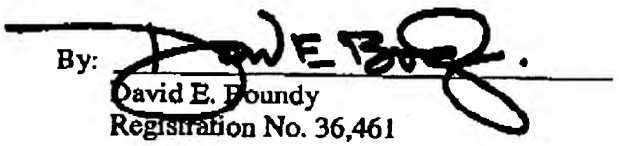
IV. Conclusion

In view of the amendments and remarks, Applicant respectfully submits that the claims are in condition for allowance, and requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. Kindly charge any additional fee, or credit any surplus, to Deposit Account No. 23-2405, Order No. 114596-28-000053BS.

Respectfully submitted,

WILLKIE FARR & GALLAGHER LLP

Dated: March 30, 2005

By: 

David E. Boundy
Registration No. 36,461

WILLKIE FARR & GALLAGHER LLP
787 Seventh Ave.
New York, New York 10019
(212) 728-8757
(212) 728-9757 Fax